REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 06-08-2013	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 6-May-2009 - 5-May-2013

4. TITLE AND SUBTITLE
Final Report: Representations and Metrics for Time-Varying Terrain Surfaces

5a. CONTRACT NUMBER
W911NF-09-1-0241

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Zachary Wartell, William Ribarsky

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of North Carolina - Charlotte
9201 University City Blvd.

Charlotte, NC 28223 -0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
55836-MA.7

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

Terrain analyses such as line-of-sight queries, trafficability, penetrability, and change feature detection are fundamental components of modern battlefield information technology. In real-time decision making, the delay in computing these analyses on higher resolution representations must be balanced against the fidelity of the calculated results. This requires time-varying terrain
models with multiple selectable resolutions and with measures of uncertainty due to data collection and data

15. SUBJECT TERMS
geo-spatial, visualization, triangle mesh, terrain modeling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Zachary Wartell
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 704-687-8442

Report Title

Final Report: Representations and Metrics for Time-Varying
Terrain Surfaces

ABSTRACT

Terrain analyses such as line-of-sight queries, trafficability, penetrability, and change feature detection are fundamental components of modern battlefield information technology. In real-time decision making, the delay in computing these analyses on higher resolution representations must be balanced against the fidelity of the calculated results. This requires time-varying terrain models with multiple selectable resolutions and with measures of uncertainty due to data collection and data approximation. The reported work makes significant contributions to research in terrain metrics and efficient computations that provide this capability. We develop several time varying data structures that incorporate terrain uncertainty measures. This includes representations for the extracted, time-varying change features and ways to access these features for meaningful exploration, understanding, and use. One interactive system uses a time-varying, multi-resolution 2D rasters and a coordinated 3D mesh. A second off-line system uses a probabilistic volumetric structure with a extractable mesh that can be incrementally augmented over time. A third interactive system integrates 3D terrain mesh visualization with a coordinated, feature-space 2D visualization.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
08/30/2012	3.00	Eric Sauda, Zachary Wartell, Jeffery Balmer, William Ribarsky. The Whole Story: Building the Complete History of a Place, 2012 45th Hawaii International Conference on System Sciences (HICSS). 2012/01/04 00:00:00, Maui, HI, USA. : ,
09/09/2011	1.00	Li Yu, Aidong Lu, William Ribarsky, Wei Chen. Automatic Animation for Time-Varying Data Visualization, Pacific Graphics 2010. 2010/09/25 00:00:00, . : ,
TOTAL:		**2**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

TOTAL:

Number of Manuscripts:

Books

Received		Paper
08/30/2012	4.00	William Ribarsky, Zachary Wartell, Wenwen Dou, Editors:, John Dill, Rae Earnshaw, David Kasik, John Vince, Pak Chung Wong. "Event Structuring as a General Approach to Building Knowledge in Time-Based Collections" in Expanding the Frontiers of Visual Analytics and Visualization, London: Springer, (04 2012)
TOTAL:	**1**	

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Isaac Cho	1.00	
Jinbo Feng	1.00	
Jialei Li	1.00	
Li Yu	0.25	
FTE Equivalent:	**3.25**	
Total Number:	**4**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Zachary Wartell	0.25	No
FTE Equivalent:	**0.25**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME PERCENT_SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See Attachment.

Technology Transfer

Final Report: Representations and Metrics for Time-Varying Terrain Surfaces

Proposal Number: 55836MA

Agreement Number: W911NF0910241

Report Period: 5/06/2009 - 5/05/2013

Authors:

Zachary Wartell, Ph.D. (PI)

William Ribarsky, Ph.D. (co-PI)

Charlotte Visualization Center

University of North Carolina at Charlotte

Abstract:

Terrain analyses such as line-of-sight queries, trafficability, penetrability, and change feature detection are fundamental components of modern battlefield information technology. In real-time decision making, the delay in computing these analyses on higher resolution representations must be balanced against the fidelity of the calculated results. This requires time-varying terrain models with multiple selectable resolutions and with measures of uncertainty due to data collection and data approximation. The reported work makes significant contributions to research in terrain metrics and efficient computations that provide this capability. We develop several time varying data structures that incorporate terrain uncertainty measures. This includes representations for the extracted, time-varying change features and ways to access these features for meaningful exploration, understanding, and use. One interactive system uses a time-varying, multi-resolution 2D rasters and a coordinated 3D mesh. A second off-line system uses a probabilistic volumetric structure with a extractable mesh that can be incrementally augmented over time. A third interactive system integrates 3D terrain mesh visualization with a coordinated, feature-space 2D visualization.

Table of Contents

Table of Figures

1 Problem Statement

The overall objectives of this 3 year project (no-cost extended to 4 years) are as follows.

O1) Develop a new compact and efficient, time-varying, hybrid mesh+volume data structure that incorporates terrain uncertainty measures

O2) Develop representations for the extracted, time-varying change features and ways to access these features as high-level events for meaningful exploration, understanding, and use.

O3) Synchronize a time-varying, multi-resolution mesh with a time-varying probabilistic volumetric structure.

O4) Create the groundwork for automated identification and classification of the change features.

O5) Develop a highly interactive, 3D user interface that allows the user to inspect and annotate the detected changes and make and incorporate corrections into the time-varying model.

The hybrid surface + volume structure will use a scalar field to record the uncertainty associated with the surface. This surface probability field (SPF) will be used to:

- combine overlapping range scans
- determine whether new incoming range scans are of higher fidelity and should replace older, lower fidelity data
- inform terrain analysis applications of the uncertainty band around the ridge surface

The surface and volume structure will both be multi-resolution and time-varying and will use a number of metrics to measure spatial and simplification error and uncertainty in the scanned data.

8The time event structure will have the following properties:

- Events will be located in space and time and will reference the underlying mesh data they correspond to.
- An event object provides a high-level concept which is far more meaningful to users than mere change features.
- Events will be determined through a combination of interaction and change detection mechanisms.

2 Summary: Year I (May 2009-July 2010)

We developed components of a probabilistic, volumetric terrain model and an interactive 3D visualization tool for terrain change detection over multiple time-steps. We also developed scripts and a workflow to acquire, clean and organize coastal LIDAR terrain data from NOAA Digital Coast. We are using 9 time steps over a 10 year span of the North Carolina coast. Extracting surfaces from a surface probability field (SPF) requires source code modifications to the marching triangles algorithm. Therefore we coded the MT algorithm with necessary hooks to switch between iso-surface and ridge surface extraction. We integrated this code with our terrain surface octree. This is a core part of our terrain model. We also developed an interactive 3D visualization tool that visualizes terrain changes between any pair of time-steps in a multi-time-step scenario. With this tool, we explored various natural and urban terrain changes in the coastal dataset. We developed data structures that link change features to their raw LIDAR data, the time interval over which the change occurred and user annotation. To allow parallel software development, the interactive tool uses a simple, fast terrain change detection method.

2.1 LIDAR Digital Coast

During Year I we acquired, cleaned and organized a LIDAR coastal terrain data set with 9 time steps over a 10 year span; we developed components of a probabilistic, volumetric terrain model; and we developed an interactive 3D visualization tool for visualizing terrain change.

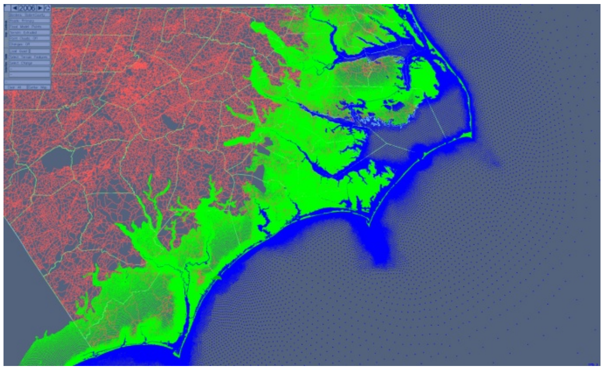

Figure 1: Our visualization tool display one year (1996) from NOAA's Digital Coast for N.C.

A long term goal of this project is to address out-of-core data processing and out-of-core visual analysis. Therefore in the summer of 2009, we assembled a large, time varying dataset. We developed scripts and C++ programs to download, translate, clean and assemble a time varying dataset from NOAA's Digital Coast. The data are LIDAR scans of the coast of North Carolina between 1996 and 2006. There 9 time steps. With our current visualization software (discussed below) we have explored a 10GB spatial subset over all years. We chose the coastal dataset because unlike the inland urban datasets we used in the past, the coastal datasets have terrain changes cause by both man-made construction as well as natural disasters such as hurricanes and flooding. Our current collection is roughly 150GB total out of 340GB available from NOAA. NOAA makes its data available through a process of requesting the data through their website and receiving a private emailed link to the requested data subset. The scripts we developed help stream-line the workflow of retrieving the data. Figure 1, illustrates the dataset from a single year. The LIDAR resolution is sufficient to capture building construction and destruction, road construction and destruction and coastal dune changes.

2.2 Marching Triangle with Implicit Surface

In Year I we also developed software components for the surface probability field (SPF) model. We must first develop a temporal-spatial hierarchical octree that encodes a scalar probability field constructed from multiple terrain scans. The individual scans are point clouds augmented with a varying probability density functions at each sample point. A scalar field is constructed by merging new scans with the current SPF model. The terrain surface is implicit in this scalar field; specifically, the terrain is the ridge surface of the scalar field.

Extracting the iso-surfaces from a scalar field is commonly performed with marching cubes or marching triangles. Marching cubes open source code is common, but when extracting ridge surfaces from multi-resolution octrees, marching triangles appears superior [1]. In particular, the look-up table needed for marching cubes assumes the grid structure is of a single resolution. Techniques for bridging the polygonalization across voxels at different resolutions generate triangles perpendicular to the actual implicit surface. This makes marching cubes less desirable than marching triangles.

Both marching triangle and marching cubes for iso-surface extraction are commonly used in closed source commercial software. However, converting an iso-surface marching algorithm to a ridge surface marching algorithm requires access and modification to the algorithm at the source code level. Source code for marching cube implementations are readily available in various packages. As of Year I (August 2009), however, open source implementations of marching triangles were unavailable or were tied to poorly extendable computational geometry libraries. Therefore, we chose to implement marching triangles with programmatic hooks to allow compile-time switching between iso-surface extraction and ridge surface extraction. We choose the CGAL C++ computational geometry library [2] due to its extensibility and generality and we coded the basic marching triangles (MT) algorithm by Hilton et al [3] [4] using CGAL.

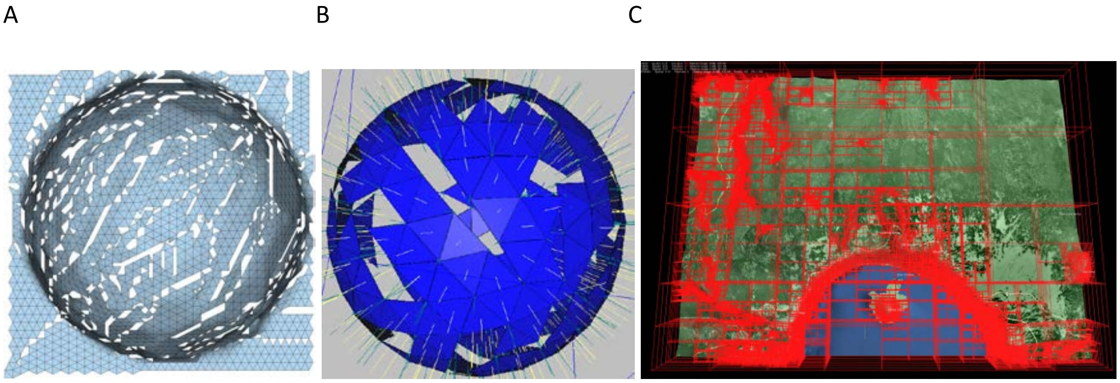

Figure 2: (A) Fournier's MT output **[5]** on an implicit analytic surface (hemi-sphere). (B) Our early MT output on an implicit analytic surface (sphere). (C) Our octree terrain surface software from earlier work which we integrated in Year II and III.

Some authors find the basic MT algorithm yields cracks for analytic surfaces. Figure 2A is from Fournier [5] where he discusses the problems and presents a set of modifications to the basic MT algorithm. One cause of cracks is the use of a fixed projection distance during the surface traversal. Possibly the crack issue may be avoided by marching through a discrete multi-resolution octree (Figure 2C) in which MT algorithm's projection distance is automatically adapted based on the current voxel size. Our simple test case for our MT code uses an analytic surface equation rather than an octree; hence our MT code

(Figure 2B) generates cracks for equation defined spheres. In prior work we developed multi-resolution octree that represents static terrain (Figure 2C) and we began refactoring and re-purposing that code.

2.3 Desktop Interactive Terrain Change Visualization Tool

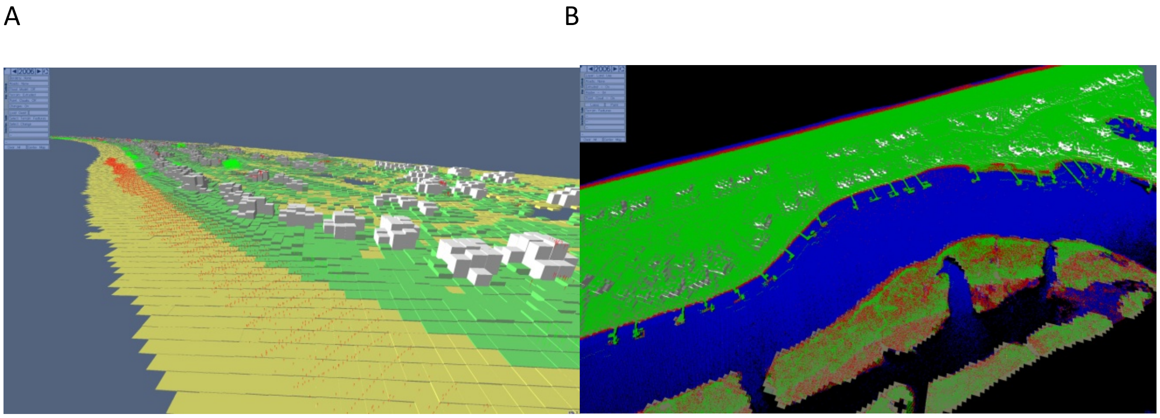

Figure 3: (A) Our visualization tool showing terrain point cloud and current 2.5D voxel representation for a strip along the N.C. coast. (B) A zoomed out view of coastal area. Blue region is water way and red and green are land. An array of boat docks are visible along the land and the small protrusions are homes.

Additionally in Year I, we developed an interactive visualization program that dynamically loads and displays user selected sub-sets of the terrain from any selected pair of the 9 time steps of the N.C. coastal data (Figure 3). We developed this visualization tool using a simple 2D image processing change detection algorithm so that we can develop our feature extraction in parallel with our SPF development. In the visualization tool, overlapping regions which have been sampled during multiple LIDAR surveys are automatically rectified and compared against each other. The change detection algorithm identifies those areas containing significant changes, and then the raw data points for each of these areas are utilized for reconstruction of a 3D change model. This change model object consists of two 3D models showing the before and after physical structures of the change, as well as measurements and characteristics extracted from the data. These measurements and characteristics are then employed within the application to classify and filter changes. We completely re-coded the user interface and internal software architecture of our earlier work [6] which only supported 2-time steps. Our new system's software supports an arbitrary number of time-steps and will support change models that are temporarily indexed. This is necessary for achieving our temporal event structure (Objective O2).

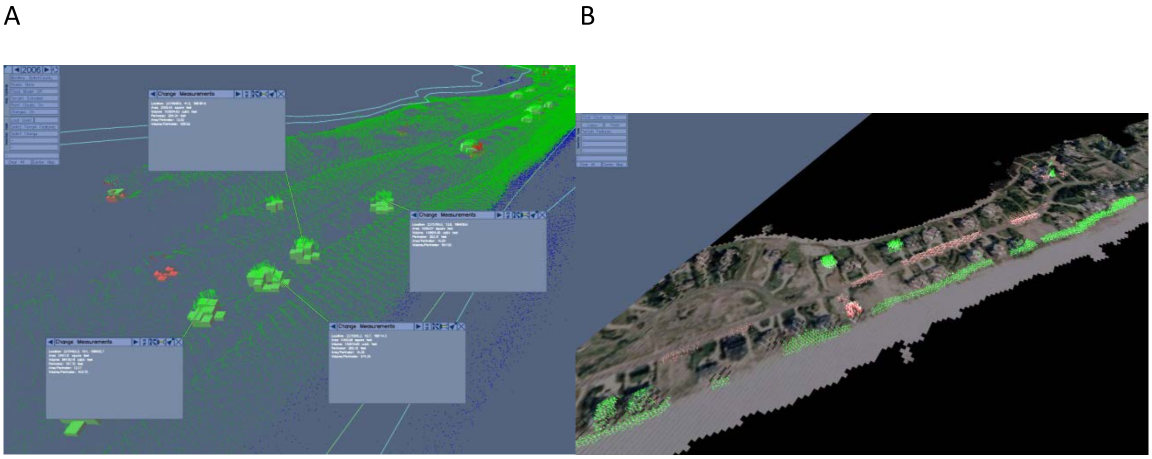

Figure 4: (A) Coastal LIDAR terrain point cloud with extracted change objects (red and green boxes) and associated probe windows for each object. (B) shows a texture map GIS layer integrated into our software.

On a common desktop workstation, the 2D change detection algorithm provides results quick enough (within a few seconds) to be done during run-time. This allows the user to not only see the results of change detection, but to interactively adjust the change-detection parameters and immediately see the effects of their adjustments. This facilitates detecting and xextracting challenging subtle features, as well as facilitating consistent detection across datasets of varying quality. In Figure 3A, the green and blue dots are the LIDAR point cloud (green is above sea-level and blue below). The green extruded raster shapes are additive changes detected by our raster change detection. These shapes correspond to newly built houses. The red extruded raster shapes are subtractive changes. Figure 3B, the terrain is rendered as an extruded raster with an overlaid aerial image. The dense green and red dots are the original LIDAR sample points overlaid on the extruded raster change objects. By controlling the 2D raster resolution, the user can generate quicker more approximate change detection and then zoom in to inspect the original LIDAR points. By supporting this 2D multi-resolution based interface, we positioned our new system's software architecture to support our in-progress multi-resolution SPF.

Figure 4A shows our integration of our latest probe based interfaces [7] [8] to the visualization tool. Each probe window is associated with one change object and displays information regarding the change object. These change objects are time-stamped and linked to the pair of terrain maps they were computed from. The objects include direct links to their underlying raw data points. These change objects and the probe interface for querying them are fundamental building blocks for the time event structure (Objective O2)).

3 Summary: Year II (August 2010 – July 2011)

During Year II, we integrated a distance-field and octree structure in our volumetric surface model and developed additional visual debugging capabilities for the marching triangles algorithm. We further developed our geo-spatial event description language and made a prototype application. Finally we identified interesting regions of terrain change in the terrain datasets that we previously assembled in Year I using our interactive tools.

3.1 Marching Triangles with Discrete Octree Distance Field

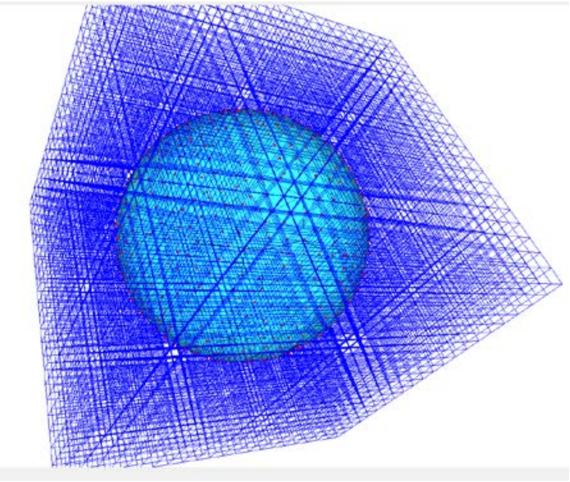

Figure 5: Octree model and visual debugging tool for volumetric probability surface model

In Year 2, we incorporated an octree encoded distance field and extended the surface extraction function to target ridge surfaces. Further, we tested a number of recent additions to the MT algorithm and incorporated some of them into our implementation. We also developed a customizable OpenGL renderer within CGAL to help visualize the executing algorithm's data structures, in particular the octree, the current triangle being considered for extension, and the neighboring ridge nodes (Figure 5).

At present, this interactive visualization tool is purely used for developing and testing our algorithms, however, the scaffolding code we implemented will also serve for future visualizations of the underlying probability field. Visualization is a key component interactive analysis. Further, developing a CGAL linked renderer is a necessary step toward being able to integrate components of our other terrain analysis applications.

3.2 Event Structure Prototype for Hurricane Storm Surge

Further in Year II, we implemented an initial event structure [9] and are further extending our initial design [10] which we published in these cited works. The event structure framework will include an event description language whose programs we refer to as *narratives*. The framework will also specify the spatial-temporal data structures and algorithms for various access methods for querying and comparing narratives. A narrative will be analogous to an execution trace of an algorithm with procedural abstraction preserved where that algorithm generates an approximation of the original dynamic dataset. We choose the trace analogy rather than an algorithm analogy because we want to computationally compare narratives. Because determining the computational equivalence of two Turing complete programs is in general uncomputable, we design the initial event description language to be a high-level description of the original dataset, rather than algorithm for generating an approximation of the dataset. There are simpler computational models, such as deterministic finite automata, for which computational equivalence is computable that we are also investigating. Figure 6 shows a hierarchical visualization of extracted events in a dynamic simulation of a hurricane storm surge from our publication [9]. Figure A illustrates a hierarchy of events (part-whole) and B shows two time-steps in the surge. The gray area is terrain while the colored region is the encroaching surge.

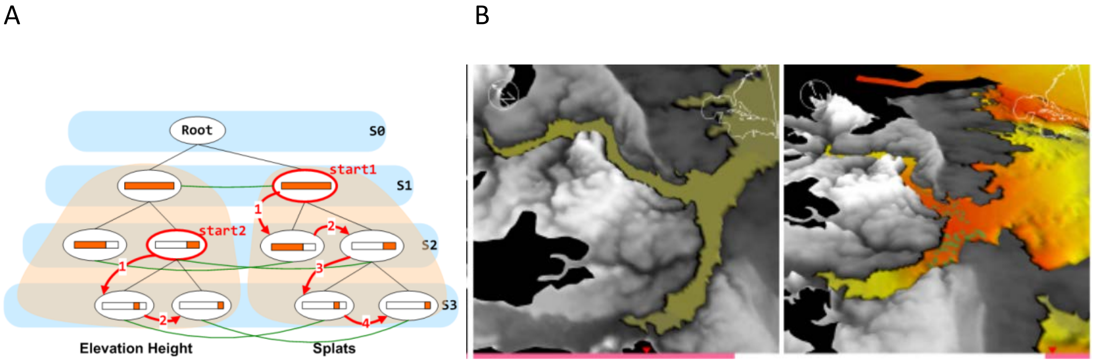

Figure 6: Event Structure with from Hurricane Surge

3.3 Application of Interactive Mesh Differencing Tool

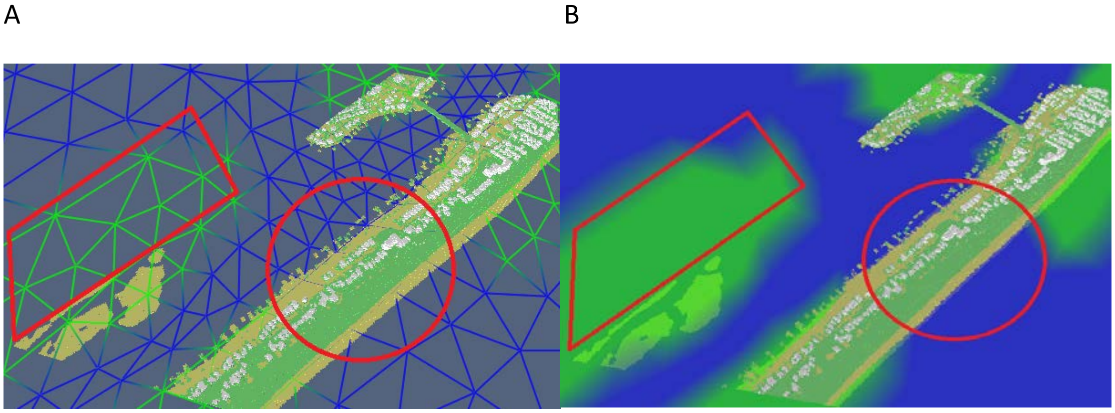

Figure 7: Differences founded between low-resolution terrain model and high-resolution terrain model.

Finally in Year II, we applied our earlier terrain surface change detection tools to find and identify regions of discrepancies between different resolutions of North Carolina coastal datasets that we assembled in Year 1. This Year 2 preliminary analysis will be useful for choosing dynamic regions of real-world data to use for our newer probabilistic algorithms. Figure 7 shows a low-resolution terrain model as wire-frame triangles while a small subset of high-resolution data is shown as solid polygons (yellow, white and green). In the low-resolution model the blue triangles are under water while the green are dry land. Note that this low-res model classifies as under water a stretch of earth that is clearly dry land in the high-res LIDAR data—this is the region inside the circle. (See also Figure 7B). Note also that the LIDAR scan does not cover the ground region inside the red quadrilateral. The discrepancy within the circle could be due to a fact-on-the-ground, i.e. during the years over which the low-resolution model was constructed perhaps the dry-land seen in the LIDAR data (circa 1998) was washed away.

4 Summary: Year III (August 2011 – July 2012)

During Year III, we progressed on several fronts. First we extended our Year 2 terrain distance field octree to a SPF (surface probability function) octree and revamped the MT algorithm as needed for extracting the terrain mesh. Next, we started a parallel development of a feature-space visual analytic tool for terrain analysis that uses a brushing-and-linking metaphor that ties 3D terrain mesh visualization to 2D statistical representations (Objective O5) and Objective O4). Finally we further developed our schema of for our proposed event structure (Objective O3) that will eventually summarize the change models extracted from the temporally evolving terrain.

4.1 Development of the Surface Probability Function (SPF) Octree

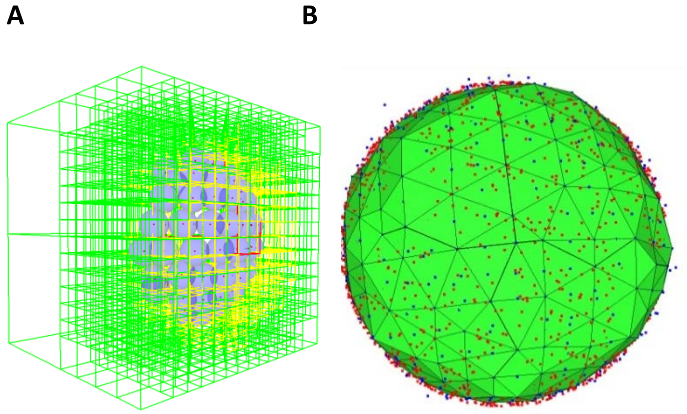

Figure 8: Visualization of our SPF Octree data structure using a sphere test surface (A) and the resulting MT surface (B).

With respect to the SPF octree, we added the Hessian, eigen-values, eigen-vector components of the SPF octree to the MT data structure. Figure 8A, shows the SPF octree for a point cloud sampling a partial sphere. The red points indicate the sample points. The transparent blue spheres are the extents of each sample point's Gaussian error shell (the radius shown is 1 standard deviation). The color coding of the octree nodes indicates the number of sample point's whose Gaussian error shell impacted the node. Green indicates the node was impacted by 8 or few sample points; yellow indicates the node was impacted by 8-25 points; and red more than 25 points. Figure 8B is the mesh extracted from the SPF octree. The original sample points are the red points. The SPF ridge points are the blue points; a subset of these are used as triangle vertices by the MT algorithm. While Figure 8B demonstrates a closed mesh, our current MT implementation is not general and requires hand-tweaking of several parameters in order to produce the closed mesh for the sphere shown.

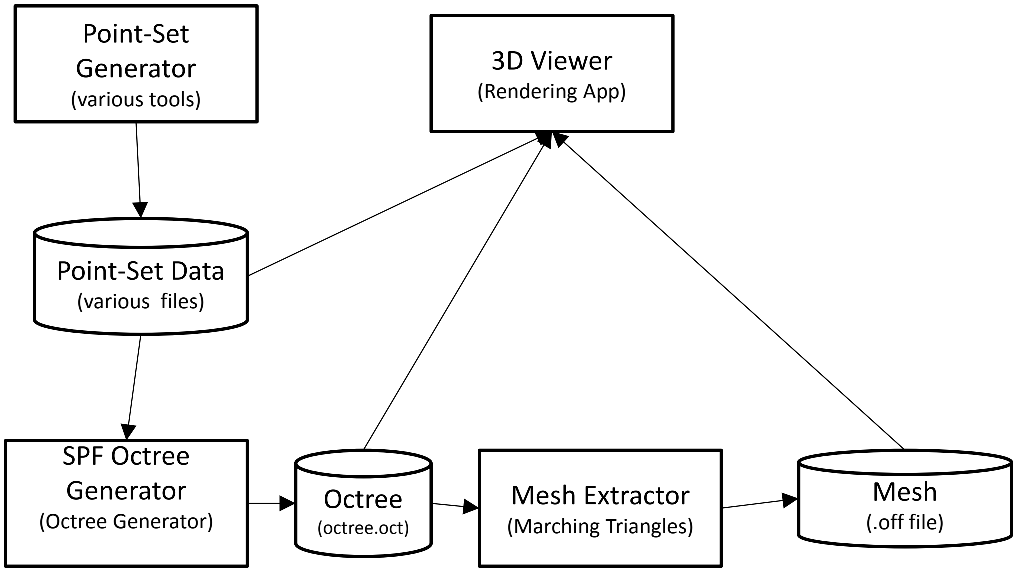

Figure 9: Flowchart diagram of our current system. Boxes indicate individual programs and cylinders are data files.

Figure 9 is a data flow diagram of our current system. The boxes are individual executable programs and the cylinders are data files. The Point-Set Generator generates simple test point clouds, such as a sampling of a sphere, and writes them to a file. The SPF (surface probability function) Octree Generator reads the point-set with per-point error information and generates an Octree file encoding the SPF. The Octree Generator also extracts the ridges points from the SPF. The Mesh Extractor module then uses a SPF modified marching triangles to extract the mesh along the SPF ridge points. The 3D Viewer can visually combine the samples points, the octree and ridge points and the final mesh.

4.2 Interactive Feature-Space Terrain Analysis Tool

Objectives O4 and O5 of this project is to use the SPF terrain model to robustly detect terrain changes and to develop an interactive, visual analytic feature-space approach for exploring and characterizing segmented terrain change models and general patches of terrain. In parallel to developing the SPF terrain software, we began to develop a terrain, feature-space tool that initially works with a simpler TIN terrain model (not a SPF based model). Our goal is to later connect this feature-space tool to the SPF terrain model. Figure 10, shows our feature-space tool. Figure 10A shows a patch of coastal terrain as a point cloud and a scatter-plot at the lower-left. The 2D plot plots one point per terrain sample point, plotting the sample's elevation and local gradient on the X and Y axis. The user selects the brown region in the plot selecting relatively low elevations with minimal gradient. This causes mostly home roofs to be high-lighted in the terrain point cloud view. In Figure 10B, the user has created various scatter-plots using different sample point statistics with the 3D point cloud in the center of the image. The user has selected different regions in each scatter plot. Color-coded 3D lines connect the selected plot points to their corresponding 3D terrain points. The aim is to provide highly interactive, multiple views of the terrain in both geospatial and abstract statistical forms.

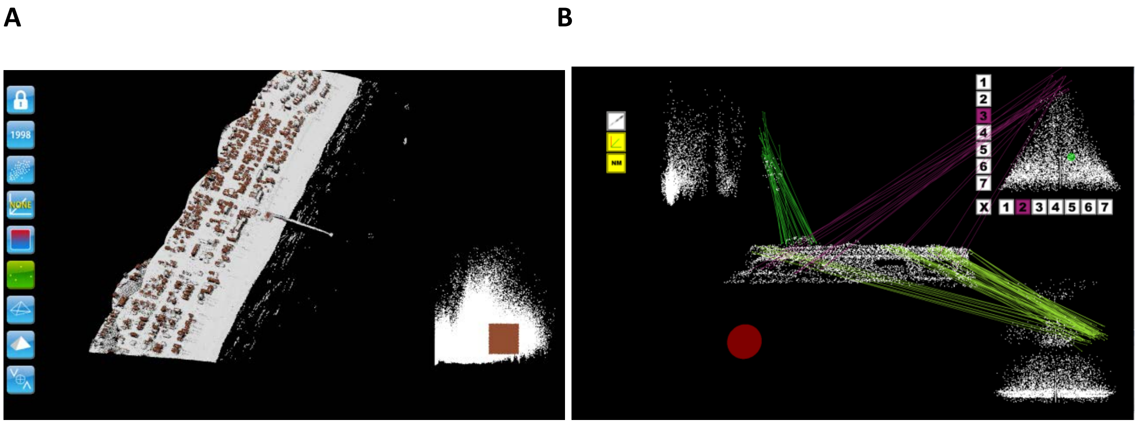

Figure 10: Linked Feature-Space and Terrain Point Cloud Visualization

4.3 Conceptual Model for Event Structure

Finally, during Year 3 we further developed our design of an object oriented schema for the event structure [10] [11] as it might be applied to historical, geo-spatial data. In our class structure, a temporal instant has 4 parts: a numeric value, a unit of measure, a calendar and a confidence descriptor. The latter three may be stored in an instant's tuple or computed. The calendar is a 1D temporal coordinate system. The confidence descriptor may indicate a confidence interval, a probability density function, or special value indicating either no error or that confidence information is not available. An instant's numeric value may be +infinite. For example, a database fact with an associated period (0,+infinite) is interpreted as holding true from instant 0 through the rest of time.

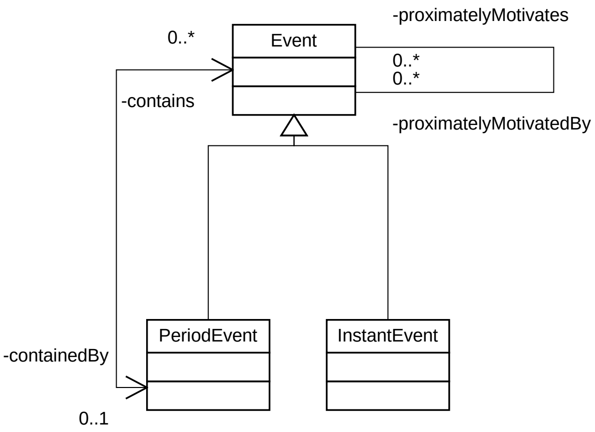

Figure 11: Abbreviated class hierarchy for our event structure

Next we define our event classes. A schematic of the class hierarchy is shown in Figure 11. An Event is an abstract class with 0 or 1 parent PeriodEvent objects (a forward declared class). PeriodEvent is an abstract class that has a valid time period and transaction time period. An InstantEvent is an abstract class that has a valid time instant and transaction time period. For example consider a sample from a digital thermometer such as (60°, 12:00PM 8/19/2011). This indicates a temperature of 60° was recorded at the given instant. A PeriodEvent also has a list of child Event objects. This parent-child structure defines a navigatable 1-to-N binary relation called TemporalContainment. The TemporalContainment relation induces a forest of tree structures on Event objects. There is a second bi-directional, navigatable N-to-N binary relation called the ProximatelyMotivates relation defined on the Event class. If Event A proximately motivates Event B, then A is an Event as a proximate cause of B. We restrict the proximatelyMotivates induced graph to be a directed-acyclic graph (DAG). Further, an Event A is said to "motivate" an event C if there is a path through the proximately motivates graph from A to C. Various constraints must be maintained between the proximately motivates and temporal containment relations to avoid semantic inconsistencies. For example, a PeriodEvent's valid time period must contain the valid time periods of all child PeriodEvent's and the instant of all child InstantEvent's.

5 Summary: Year IV: (August 2012– May 2013)

We were granted a no-cost extension for Year 4. This year we submitted several publications [12] [13] on the terrain analysis tools developed and discussed earlier (Figure 3, Figure 7, Figure 10). We explored SIFT based terrain feature analysis (an aspect of Objective O5) and further developed the SPF terrain model. While the latter is as yet incomplete it remains key to the primary thesis research topic of Ph.D. student Jialei Li, who was a research assistant on this grant.

5.1 SIFT Feature Analysis

In Year IV, we experimented using SIFT (Scale-Invariant Feature Transform) as a terrain analysis tool. Two major uses of feature detection are (1) to find features in terrain and match and track them across different years and (2) allow the user to manually select features in one terrain patch and have an algorithm find similar features in other locations. We use data from the NC Coastal database. We convert the terrain point clouds to a height-field raster, displayable as a gray-scale image. We first investigated using a 3D, volumetric variant of SIFT which stacks the time varying terrain images using time as a third dimension. Colleagues had shown this technique useful for time-varying volumetric data [14]. Our theoretical evaluation, however, showed that 3D SIFT would not be able to find space-time features due to the gross asymmetry between the width and height of the geospatial dimensions compared to the available "depth" of the temporal dimension. Therefore, we pursued applying 2D SIFT to each snapshot and then correlating the features across years.

We tested SIFT on pairs of years of several terrain patches. The goal is to extract and track the appearance and disappearance of geometric terrain features between years. Figure 12A and B (next page) show one patch of coastal terrain over two successive years. The arrows indicate the location and major orientation of the SIFT features with the longer arrows corresponding to larger features (of lower spatial frequency). Notably many features appear in only one year and not the other; for instance the large circled features shown in B are not found in A. Gaining any intuitive understanding of what is the significance of these features proved quite difficult. For example, examining Figure 12A and B and considering the largest circled arrow, it is not clear which large, low resolution component in B this arrow corresponds to which does not also occur in figure A. Manually exploring the Gaussian image pyramid was not particularly illuminating. Possibly exploring the full set of intermediate difference of Gaussians (DoGs) would be more insightful; however based on our manual exploration it is unclear whether even a good interactive tool for doing so would yield physically meaningful results.

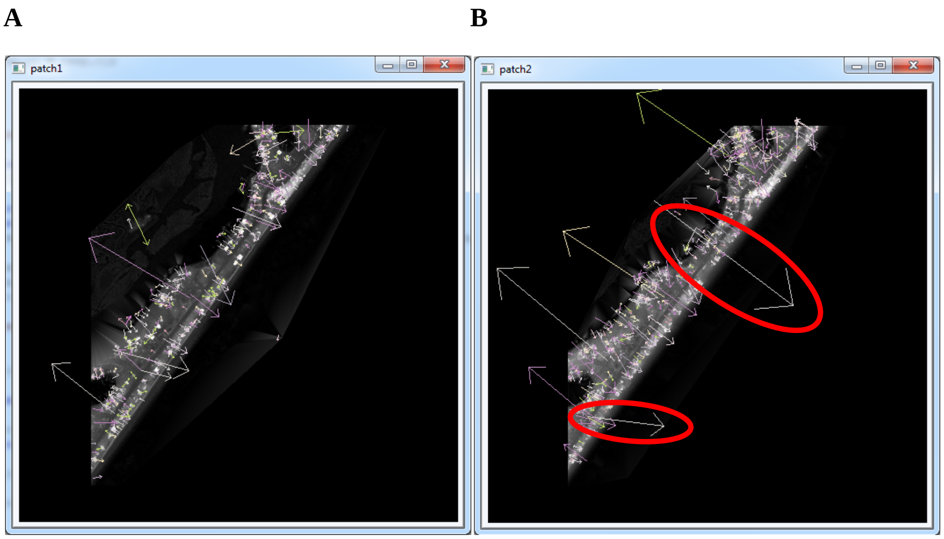

Figure 12: Extracted SIFT 2D features in two years of a patch of terrain.

A second use of terrain feature detection is to allow the user to select features in one area based on a hypothesis such as whether the area's local features are correlated with a geospatial event, such as flooding, and then to automatically select similar features in other locations in order to explore whether similar events occur at these matching locations.

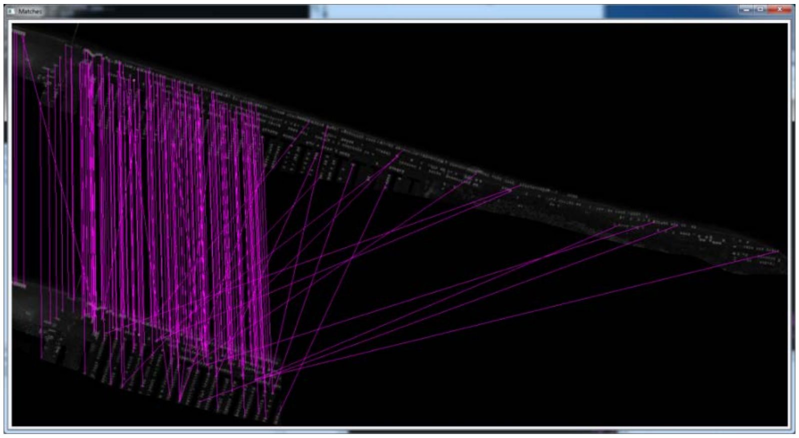

Figure 13: Feature matching between a separately processed terrain patch (bottom half of window) and the same patch when processed as part of a group of three registered patches (top half of window). Purple lines connect matching features.

As a preliminary step, we explored to what degree the features in a single patch would also be found in adjacent patches. Figure 13 shows three patches of terrain aligned and registered together in the top half of the window and then the left most constituent patch repeated separately in the bottom half of the image. (The terrain images show rows of homes along a beach front). Lines connect matching features found in the joined 3-set (top) against features found when processing the right most patch separately (bottom). Most of the lines simply indicate the algorithm made the obvious matches between the separately processed patch and the same part within the 3-set. However, some of the lines indicate that some features have matches in adjacent patches. Manual inspecting these matches, we only found house size features being matched between patches and not large, low-frequency features. This seems to limit the usefulness of applying SIFT. For instance, it demonstrates that a SIFT feature could not capture the signature of a larger area of a certain roughness characteristic because this signature involves both high-frequency variation and a large area while standard SIFT finds features whose spatial frequency component is inversely related to its areal extent.

5.2 SPF Terrain Model

In Year 4, we continued to develop the SPF processing algorithms in our pipeline (Figure 14, next page) and began testing the pipeline on terrain patches. Figure 14A shows the original sample points (red) of a patch of terrain with a higher-resolution inset. Figure 14B is a Delanauy triangulation of the entire set of sample points without any consideration of sample error. Figure 14C shows the sample points with their Gaussian error shells. Figure 14D shows one slice of the SPF octree after the sample points are inserted. The dark blue points are a subset of the SPF octree's ridge points. A slice of octree nodes are displayed as wireframe, light blue boxes and several levels are shown. Figure 14E shows both the original sample points and the entire set of SPF ridge points color coded based on whether the ridge point is classified as an outlier by the SPF algorithm. The blue are non-outlier ridge points while other colors distinguish several classes of outliers. The image demonstrates that the average density of ridge points is similar to the average density of sample points. Figure 14F shows the preserved ridge points (i.e. with the outliers removed) that are considered by the MT algorithm. This image demonstrates that the ridge points fall into a more regular grid pattern than the original sample points and this pattern is aligned with the leaf nodes. Figure 14G shows these ridge points along with the leaf level octree nodes that contain them and with purple lines indicating the nodes' principle curvature vectors and blue line the SPF gradient vectors.

Our current SPF MT implementation still generates some holes. Figure 14H illustrates the current result. In Figure 14I, triangles are color coded based on the MT step [3] which added them: dark blue is a 'new' triangle, light blue is new triangle that connected with a previously added vertex (this is SPF specific), yellow is a 'previous' triangle, purple is 'next' triangle, red is an 'overlap' triangle, and black is the SPF seed triangle. We have experimented with a dozen combinations of the MT algorithm modifications suggested by Fournier [5]. Many turn out not to be applicable to SPF octrees. The remaining mesh cracks will be filled using Contour Meshing from Akkouche and Galin [15]. We are fixing the ridge point gaps and will perform further testing. We are writing a manuscript regarding combining the Fournier and Akkouche *et al.* MT additions with the SPF Octree of Johnson and Manduchi [1]. Jialei Li, a Ph.D. student and research assistant under this contract, will be presenting this manuscript for his Qualifying Exam this Fall.

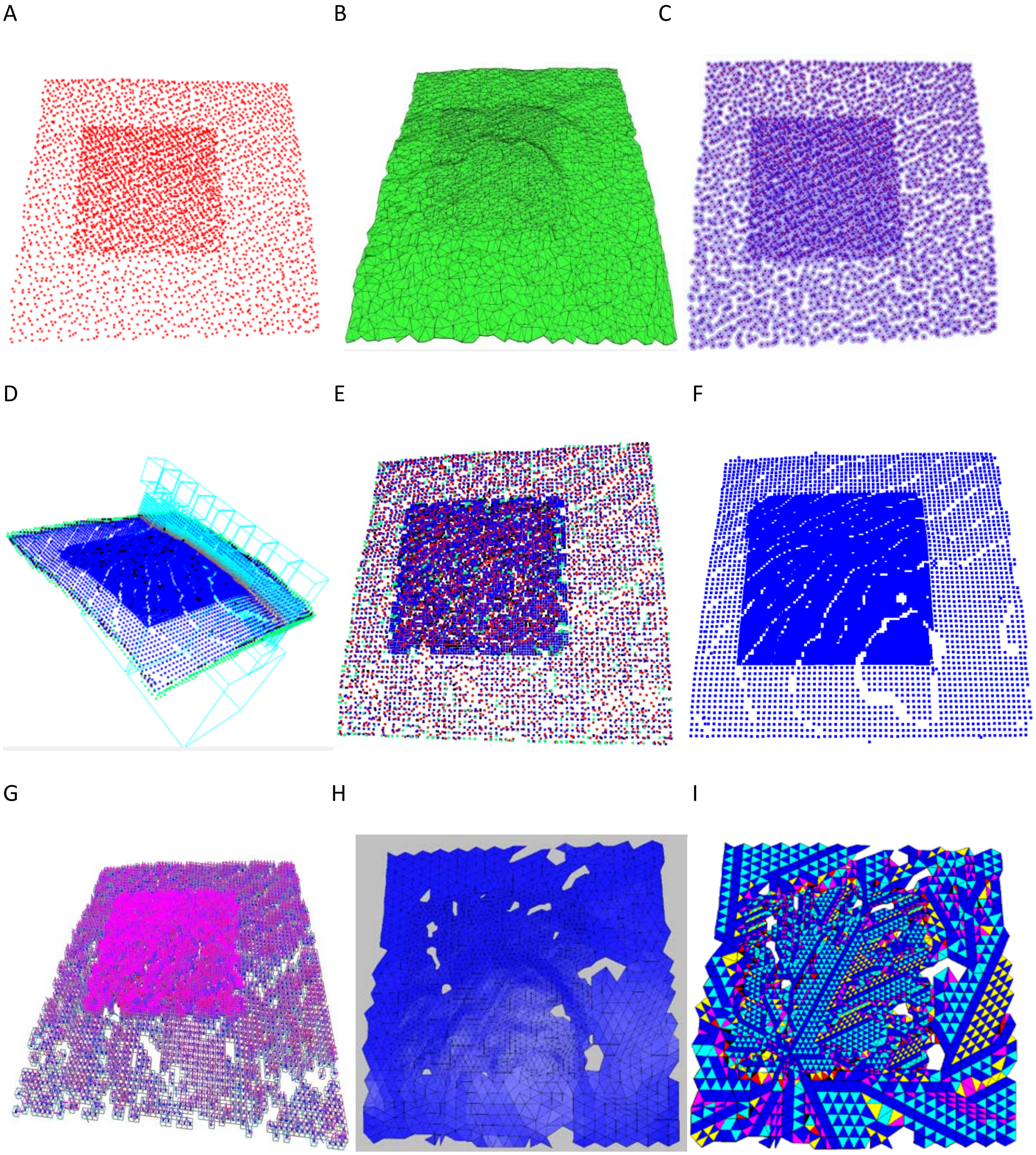

Figure 14: (A) Sample Points. (B) Delaunay Triangulation of Sample Points. (C) Sample Points with Gaussian Error Boundaries. (D) Slice of SPF Octree and Ridge Points. (E) Ridge Points (including outliers) and Sample Points. (F) Ridge Points (outliers removed). (G) Octree Leaf Nodes with Gradient Vectors. (H) Current SPF Octree Marching Triangles output (color coded, see text). (I) Current SPF MT output with lighting

6 Conclusion and Future Work

We achieved many aspects of most of our objectives. Publications [11] [9] [10] present our work on the concept of event structuring for geo-spatial data (Objective O2). Submitted publications [12] [13] describe our interactive 3D terrain visual analysis tools illustrated in Figure 3, Figure 4, Figure 7 and Figure 10 (Objective O5 and O4). These utilize a much simpler terrain model than targeted by Objective O1. This was required in order to develop the different components in parallel. Implementing the SPF Octree remains a challenge but a manuscript is pending. We did not make as much progress as desired on Objective O1. Objective O3 requires tying the SPF Octree of Objective O1 to more sophisticated terrain rendering. Objective O3 was the most elusive because it required completing the SPF Octree first and then integrating it with efficient rendering techniques and bringing that technology into our interactive tools for terrain analysis. Our immediate goal is to complete the SPF Octree implementation and evaluate it with more varied datasets. In parallel we will also integrate the SPF Octree into the newer 3D analysis tool with linked-feature spaces (Figure 10).

7 Bibliography

[1] A. E. Johnson and R. Manduchi, "Probabilistic 3D Data Fusion for Adaptive Resolution Surface Generation," Los Alamitos, CA, USA, 2002.

[2] *CGAL, Computational Geometry Algorithms Library.*

[3] A. Hilton, A. J. Stoddart, J. Illingworth and T. Windeatt, "Implicit surface-based geometric fusion," *Comput. Vis. Image Underst.,* vol. 69, no. 3, pp. 273-291, 1998.

[4] A. Hilton, A. Stoddart, J. Illingworth and T. Windeatt, "Marching triangles: range image fusion for complex object modelling," *Image Processing, 1996. Proceedings., International Conference on,* vol. 1, pp. 381-384 vol.2, Sep 1996.

[5] M. Fournier, "Surface Reconstruction: An Improved Marching Triangle Algorithm for Scalar and Vector Implicit Field Representations," *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images,* vol. 0, pp. 72-79, 2009.

[6] T. Butkiewicz, R. Chang, W. Ribarsky and Z. Wartell, "Understanding Dynamics of Geographic Domains," K. S. H. May Yuan, Ed., CRC Press/Taylor and Francis, 2007.

[7] T. Butkiewicz, R. Chang, Z. Wartell and W. Ribarsky, "Visual analysis and semantic exploration of urban LIDAR change detection," in *Proceedings of the 10th Joint Eurographics / IEEE - VGTC conference on Visualization*, Aire-la-Ville, Switzerland, Switzerland, 2008.

[8] T. Butkiewicz, R. K. Meentemeyer, D. A. Shoemaker, R. Chang, Z. Wartell and W. Ribarsky, "Alleviating the Modifiable Areal Unit Problem within Probe-Based Geospatial Analyses," *Computer*

Graphics Forum, vol. 29, no. 3, pp. 923-932, 2010.

[9] L. Yu, A. Lu, W. Ribarsky and W. Chen, "Automatic Animation for Time-Varying Data Visualization," in *Computer Graphics Forum*, 2010.

[10] W. Ribarsky, Z. Wartell and W. Dou, "Event Structuring as a General Approach to Building Knowledge in Time-Based Collections," in *Expanding the Frontiers of Visual Analytics and Visualization*, J. Dill, R. Earnshaw, D. Kasik, J. Vince and P. C. Wong, Eds., Springer London, 2012, pp. 149-162.

[11] W. Ribarsky, E. Sauda, Z. Wartell and J. Balmer, "The Whole Story: Building the Complete History of a Place," in *45th Hawaii International Conference on System Sciences*, 2012.

[12] X. Wang, I. Cho, T. Butkiewicz and Z. Wartell, "Towards Utilizing Heterogeneous Analytics Interfaces in Coastal Infrastructure Management," in *Submitted to IEEE Pacific Vis 2013 [manuscript electronically available as ARO Tech Report]*, 2013.

[13] I. Cho, X. Wang and Z. Wartell, "HyFinBall: A Two-Handed, Hybrid 2D/3D Desktop VR Interface for Visualization," in *Submitted to IEEE Vis 2013 [manuscript available eletronically as ARO Tech Report]*, 2013.

[14] L. Yu, "Abstract Visualization of Large-scale Time-varying Data," Ph.D. Dissertation, Dept. of Computer Science, University of North Carolina at Charlotte, Charlotte, NC , 2012.

[15] S. Akkouche and E. Galin, "Adaptive implicit surface polygonization using marching triangles," in *Computer Graphics Forum*, 2001.

[16] T. Chen, A. Lu and S.-M. Hu, "Visual storylines: Semantic visualization of movie sequence," *Computers \& Graphics,* vol. 36, no. 4, pp. 241-249, 2012.

[17] W. Ribarsky, E. Sauda, Z. Wartell and J. Balmer, "The Whole Story: Building the Complete History of a Place," in *45th Hawaii International Conference on System Sciences*, 2012.